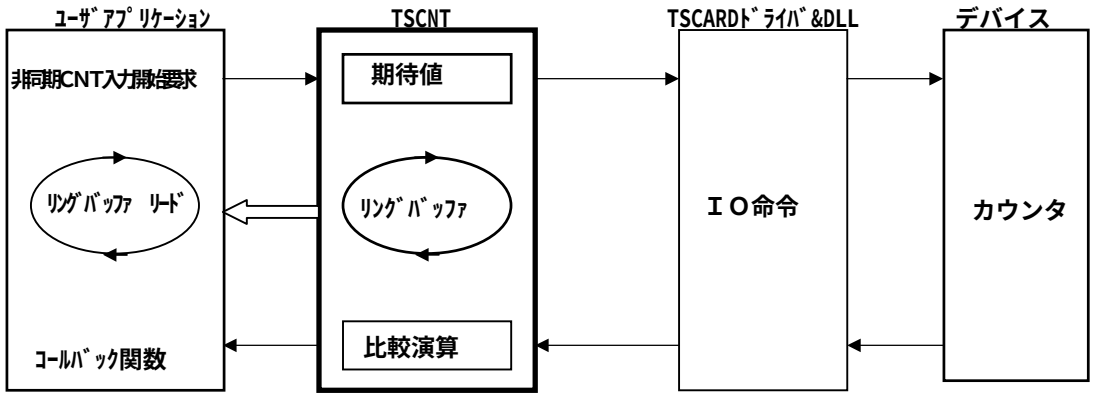
ＴＳＣＮＴ

概要

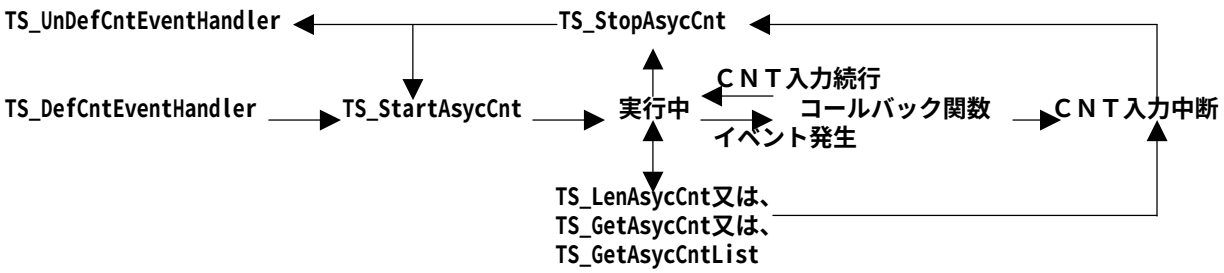


非同期カウンタ入力要求は同一デバイスに対して、チャネル毎に独立して要求することが出来ます。
非同期カウンタ入力要求があるとＤＬＬ内部でチャネル毎のスレッドを生成します。このスレッドにより指定された周期でカウンタ入力を行い、結果をリングバッファに格納します。リングバッファはカウンタ入力要求時ＤＬＬにより動的に確保します。そのサイズは1チャネル当たり最大３２メガポイント（３２ビット）まで指定可能です。リングバッファがフルになるとカウンタ入力結果のワードにその旨のフラグをセットし前データに上書きします。
非同期カウンタ入力の他に下記の機能があります。

・カウンタ入力値と期待値との比較演算を行い、条件が成立するとコールバック関数を実行します。
・リングバッファフルによりコールバック関数を実行します。

制御手順

制御に先立ってデバイスをオープンしてください。また終了時はデバイスをクローズしてください。
１．TS_DefCntEventHandler 関数でコールバック関数を登録してください。
２．TS_StartAsycCnt 関数でＣＮＴ入力を開始してください。
　・AD入力実行中イベント発生条件が成立するとコールバック関数が呼び出されます。
　・コールバック関数実行後ＣＮＴ入力を続行するか否かは指定によります。尚、ＣＮＴ入力中断中でも TS_LenAsycCnt、TS_GetAsycCnt、TS_GetAsycCntList は実行可能です。
　・コールバック関数の実行は一度のみです。再度イベントを監視したい場合は TS_StopAsycCnt 実行後 TS_StartAsycCnt を行ってください。
３．TS_StopAsycCnt 関数でＣＮＴ入力は停止します。停止後再度ＣＮＴ入力したい場合は 2 項に戻ってください。
４．TS_UnDefCntEventHandler 関数でコールバック関数を解除してください。



ＡＰＩ関数

●コールバック関数

int WINAPI TS_CntEventHandler(HANDLE DeviceHandle,DWORD ch,DWORD lParam);

【引数】
 HANDLE DeviceHandle　　:デバイスハンドル
 DWORD ch　　:チャネル番号
 DWORD lParam　　:ユーザ定義メッセージの付加情報
　非同期ＣＮＴ開始時、ＣＮＴパラメータ構造体で指定したlParamメンバです。
【戻り値】
 int　　:正常終了の場合は０、異常終了の場合は負の値を返します。

【説明】
 ユーザアプリケーション内に記述するコールバック関数です。
 イベントタイプで指定された条件が成立したとき本関数をコールバックします。
 【参照⇒ＣＮＴパラメータ構造体】
 尚、TS_CntEventHandler　はプレースホルダーであり、実際にこの関数名を使う必要はありません。

【Visual Basicでの記述】

```
Public Function CntEventHandler(ByVal DeviceHandle As Long,  ByVal ch As Long, _
                                ByVal LParam As Long) As Long
```

●コールバック関数の登録

int WINAPI TS_DefCntEventHandler (IN HWND hwnd, IN USERHANDLER CntEventHandler);

【引数】
 IN HWND hwnd　　:ウィンドウハンドルまたはＮＵＬＬ。
 IN USERHANDLER CntEventHandler　　:コールバック関数アドレス

【戻り値】
 int　　:正常終了の場合は０、異常終了の場合は負の値を返します。

【説明】
 コールバック関数を登録します。
 ウィンドウハンドルは、ＧＵＩウィンドウから呼び出す場合そのウィンドウハンドルを指定してください。
 コンソールアプリケーションから呼び出す場合ＮＵＬＬを指定してください。
【参照⇒使用上の注意：コールバック関数の注意事項】

【Visual Basicでの記述】

```
Public Declare Function TS_DefCntEventHandler Lib "TSCNT" (ByVal hwnd As Long, _
                                                    ByVal CntEventHandler As Long) As Long
```

●コールバック関数の解除

int WINAPI TS_UnDefCntEventHandler (void);

【引数】
 なし。

【戻り値】
 int　　:正常終了の場合は０、異常終了の場合は負の値を返します。

【説明】
 全てのチャネルを停止後、コールバック関数の登録を解除します。

【Visual Basicでの記述】

```
Public Declare Function TS_UnDefCntEventHandler Lib "TSCNT" () As Long
```

●非同期ＣＮＴ開始
int WINAPI TS_StartAsycCnt (IN HANDLE DeviceHandle, IN TS_CNTREQBLOCK *reqblock);

【引数】
 HANDLE DeviceHandle　　:デバイスハンドル。
 TS_CNTREQBLOCK *reqblock　:ＣＮＴパラメータ構造体へのポインタ。

【戻り値】
 int　　:正常終了の場合は０、異常終了の場合は負の値を返します。

【説明】
 関数呼び出しに成功すると、関数内部でチャネル毎のスレッドを生成します。
 このスレッドは指定されたチャネルを周期的に読み出しリングバッファに格納します。
 一方イベントタイプで指定された事象を監視し、条件が成立するとコールバック関数を呼び出します。
【参照⇒使用上の注意：コールバック関数の注意事項】

 コールバック関数呼び出しは一度のみですが、ＣＮＴ入力は継続または中断します。（指定による）

【Visual Basicでの記述】
```
Public Declare Function TS_StartAsycCnt Lib "TSCNT" (ByVal DeviceHandle As Long, _
                                                     ByRef reqblock As TS_CNTREQBLOCK) As Long
```

●非同期ＣＮＴ停止
int WINAPI TS_StopAsycCnt (IN HANDLE DeviceHandle, IN DWORD ch);

【引数】
 HANDLE DeviceHandle　　:デバイスハンドル。
 DWORD ch　　:チャネル番号

【戻り値】
 int　　:正常終了の場合は０、異常終了の場合は負の値を返します。

【説明】
 チャネル番号で指定された非同期ＣＮＴ入力を停止し、リングバッファを開放します。チャネル番号が－１の場合
 全てのチャネルを停止します。

【Visual Basicでの記述】
```
Public Declare Function TS_StopAsycCnt Lib "TSCNT" (ByVal DeviceHandle As Long, _
                                                    ByVal ch As Long) As Long
```

●非同期ＣＮＴリストの取得
int WINAPI TS_GetAsycCntList(IN HANDLE DeviceHandle, IN DWORD ch, OUT TS_CNTREQBLOCK *reqblock);

【引数】
 HANDLE DeviceHandle　　:デバイスハンドル。
 DWORD ch　　:チャネル番号
 TS_CNTREQBLOCK *reqblock　:ＣＮＴパラメータ構造体へのポインタ。

【戻り値】
 int　　:正常終了の場合は０、異常終了の場合は負の値を返します。

【説明】
 チャネル番号で指定されたＣＮＴパラメータ構造体を返します。

【Visual Basicでの記述】
```
Public Declare Function TS_GetAsycCntList Lib "TSCNT" (ByVal DeviceHandle As Long, _
                                                       ByVal ch As Long, _
                                                       ByRef reqblock As TS_CNTREQBLOCK) As Long
```

●非同期ＣＮＴの取得

int WINAPI TS_GetAsycCnt(IN HANDLE DeviceHandle, IN DWORD ch, OUT DWORD *data);

【引数】

HANDLE DeviceHandle	:デバイスハンドル。
DWORD ch	:チャネル番号。
WORD *data	:ＣＮＴ入力データポインタ。【参照⇒入力データフォーマット】

【戻り値】

int :正常終了の場合はリングバッファのデータ数、異常終了の場合は負の値を返します。

【説明】

チャネル番号で指定されたリングバッファリードします。

【Visual Basicでの記述】

```
Public Declare Function TS_GetAsycCnt Lib "TSCNT" (ByVal DeviceHandle As Long, _
                                                   ByVal ch As Long, _
                                                   ByRef data As Long) As Long
```

●非同期ＣＮＴデータ数の取得

int WINAPI TS_LenAsycCnt(IN HANDLE DeviceHandle, IN DWORD ch, OUT DWORD *buflen);

【引数】

HANDLE DeviceHandle	:デバイスハンドル。
DWORD ch	:チャネル番号。
DWORD *buflen	:データ数格納アドレス。

【戻り値】

int :正常終了の場合は０、異常終了の場合は負の値を返します。

【説明】

チャネル番号で指定されたリングバッファのデータ数をリードします。

【Visual Basicでの記述】

```
Public Declare Function TS_LenAsycCnt Lib "TSCNT" (ByVal DeviceHandle As Long, _
                                                   ByVal ch As Long, _
                                                   ByRef buflen As Long) As Long
```

入力データフォーマット

bit0～23　　ＣＮＴ値。リードレジスタをバイナリ変換した値です。【参照⇒取扱説明書】
bit30　　未使用。
bit31　　1: リングバッファ上書きデータです。

ＣＮＴパラメータ構造体

(o)印はＤＬＬが格納する情報です。

```
typedef struct {
DWORD   ch;       チャネル(0～3)
DWORD   cntmode;  モード（0～1)
DWORD   bufsize;  リングバッファサイズ(1～32ﾒｶﾞﾎﾟｲﾝﾄ)
DWORD   time;     ＣＮＴ周期(0～65535ms)
DWORD   lParam;   ユーザ定義メッセージの付加情報
DWORD   type;     イベントタイプ
DWORD   mask;     マスク(0x0～0x01ffffff)
DWORD   expec;    期待値（符号付３２ビット：-16777216～+16777215）
(o)DWORD  evval;  イベント値
(o)DWORD  evcause; イベント発生原因
}TS_CNTREQBLOCK,*PTS_CNTREQBLOCK;
```

ch　　チャネル番号【参照⇒取扱説明書】
　　0:CH1,1:CH2,2:CH3,3:CH4
cntmode　モード【参照⇒取扱説明書】
　　0:HEXﾓｰﾄﾞ,1:BCDﾓｰﾄﾞ
bufsize　リングバッファサイズ
time　　ＣＮＴ周期【参照⇒使用上の注意：入出力周期について】
lParam　アプリケーションが任意に使用できるパラメータです。コールバック関数のlParamと同じ値です。
　　（使用例）イベント発生までの時間計測などに使用できます。
　　非同期ＣＮＴ開始要求時に本パラメータに現在時刻を設定しておき、イベント発生時コールバック関数内でこの時刻と現在時刻との差を求めます。この差を計測値として使用します。
type　　イベントタイプを指定します。各ビットは同時指定可能です。
　　bit0～7　　0:比較演算無し。（注１）
　　1:（ＣＮＴ値　and　マスク）＝期待値　の時コールバック関数実行。
　　2:（ＣＮＴ値　and　マスク）＞期待値　の時コールバック関数実行。
　　3:（ＣＮＴ値　and　マスク）＜期待値　の時コールバック関数実行。
　　bit8　　1:リングバッファフルの時コールバック関数実行。
　　bit15　　0: コールバック関数実行後ＣＮＴ入力続行。
　　1: コールバック関数実行後ＣＮＴ入力中断。
　　（注１）比較演算はＣＮＴ値の符号を拡張し、符号付３２ビットで行います。
mask　　比較演算の有効ビットを本値でマスクします。
expec　　比較演算の時の期待値です。
evval　　イベント発生時の入力データが格納されます。
evcause　typeフィールドで示されたイベント発生原因が格納されます。

パラメータ例

チャネル１において、リングバッファフルに先だってＣＮＴ値（HEXﾓｰﾄﾞ）の全ビット（mask:0x01ffffff）が<expec:100>を超えた時コールバック関数を実行します。このとき、<evval:101>、<evcause:0x0002>が格納されます。またコールバック関数実行後ＣＮＴ入力は中断はします。

ch	0	チャネル１
cntmode	0	HEXﾓｰﾄﾞ
bufsize	10000	リングバッファサイズ(10kﾎﾟｲﾝﾄ)
time	1	ＣＮＴ周期(1ms)
lParam	0	ユーザ定義メッセージの付加情報
type	0x8102	リングバッファフル、又は（ＣＮＴ値andマスク）＞期待値の時コールバック関数実行
mask	0x01ffffff	マスク
expec	100	期待値
evval	0→101	イベント値
evcause	0→0x0002	イベント発生原因

異常終了ステータス

-128：ＣＮＴ入力停止中である。
-129：ＣＮＴ入力パラメータエラー。
-130：ＣＮＴ入力開始失敗
-131：ＣＮＴ入力実行中である。
-132：未使用。
-133：リングバッファ確保不可。
-134：コールバック関数登録または解除失敗。
-135：コールバック関数未登録である。